40205163 Coursework Report

Calum Hamilton
40205163@napier.ac.uk
Edinburgh Napier University - Module Title (SET008101)

Abstract

Coursework Brief The coursework given was designed to show that we had acquired a number of skills over the term of the course: an understanding of the World Wide Web, a solid understanding of web page development, an accrual of skills working with different web authoring tools and languages and the ability to combine a set of web pages and a web server to produce a working product.

Report Summary The report below is designed to describe the process I took from starting the project to the end product, and an analysis of each of the stages. It will also look to provide details of the learning process undertaken throughout the entire project. The coursework files will be in the same private repository with the report I have produced. The website has been produced specifically for the coursework project and is not planned to be used in any later applications. The report and coursework have only been shared with Simon Wells and myself at the time of writing.

Keywords – HTML, CSS, JavaScript, Report, Coursework, University, Web, Tech, Coursework, Napier

1   Introduction

Assignment The assignment given was to create a simple blog platform, using server and client-side coding. This is a leap from the last coursework since it was only client-side code that we needed to produce in the first coursework. The basic functionality of the blog posts include adding, editing and deleting blog posts with data persistence. There was also a lot of scope for extension in this coursework, looking for us to push our abilities, creativity and learning. This coursework is also quite a lot more advanced than the last coursework, as data persistence and server-side coding is something that we weren't required to do in the first coursework.

Website Features I created a simple 4 page website: a login page, blog home page, a page to edit the blog and a page to create a new blog post. The navigation system is through a navbar, with an unimplemented search bar. The navbar was designed to provide functionality to easily access the commonly used parts of the site, while the search bar was designed to search for other users within the blog and bring up their blog page, with all their posts. The search bar was required for finding other blogs, since

the dataset of names is too large for any other display style.

Website Functionality Of the HTML features I have implemented, the sign up, login and the search functions don't work. I will analyze these complications later in the report. The login and sign up functions both open up a menu which allows the user to enter the relevant data. The edit profile page has a set of fields where you can actively update the details of your profile page, including a bio, age and occupation. The new post page has a field for title entry and a post, which both get stored and displayed under the user profile. The profile displays all the information, along with a profile picture and options for updating or deleting current posts.

2   Software Design

Approach I wanted to change up my approach from the last website that I made for the coursework. I firstly went to the library and loaned out the "Mobile Web Designer's Idea Book" and "The Web Designer's Idea Book - Volume 3". I read a number of books and articles online about web development to strengthen my understanding before I attempted the challenging project. I drew heavy inspiration from my previous coursework for the style and I went back to the idea of a standard navbar and a centered website as the style looks clean, although to alternate, I removed a number of pictures as I found them to cluster up the page. After also spotting a number of other standard features from blog platforms in the Web and Mobile Web Designer Books, I then went to Facebook and Twitter to check if there were other features to implement. The age, gender and occupation came as a result of that, although I left out relationship status, as I'd rather it remain a blog site and not a dating site.

For a colour scheme, I went back to "https://coolors.co/" and looked for a clean and unique colour scheme. I found a very attractive purple I liked and the yellow and pink seemed to fit the colour scheme. I then relied on "https://wireframe.cc/" to get a basic layout in place, however unlike last time, I also drew up a large number of plans and kept sketching the pages for my site and a quick layout. I wanted to ensure that I was very happy with my layout and colour scheme before I even started coding.

I kept all the pages at the same level, although had the idea to redirect users to the login page if they weren't already logged in if accessing a custom part of the site,

which I attempted to implement. I didn't see a need for a hierarchy in such a small website, since all pages were on the same navbar. If I had other pages I would have looked at creating sub menus through a dropdown menu.

3 Implementation

I worked in Atom for creating the web application. I started with the index login page to get the basic colour scheme finalized and to get a base to build the best of the site from. I made several dummy html pages to test out all the elements and all of the elements I wanted to try. [1][2][3][4]
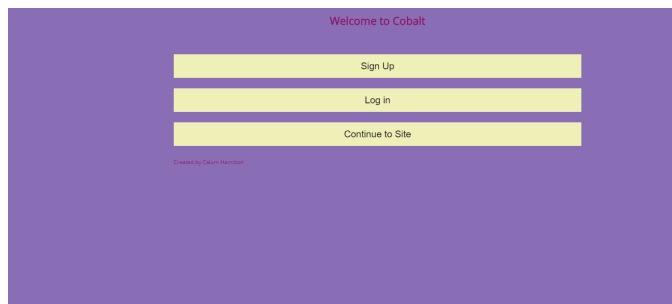

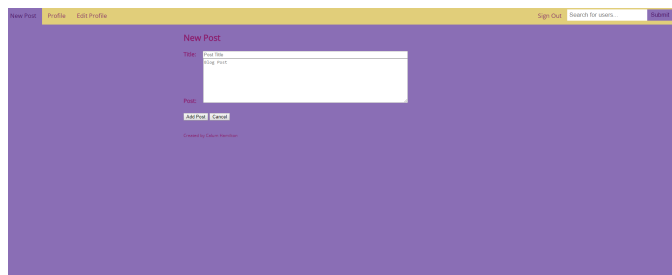Figure 1: **Webpage Screenshot** -Screenshot of the Index

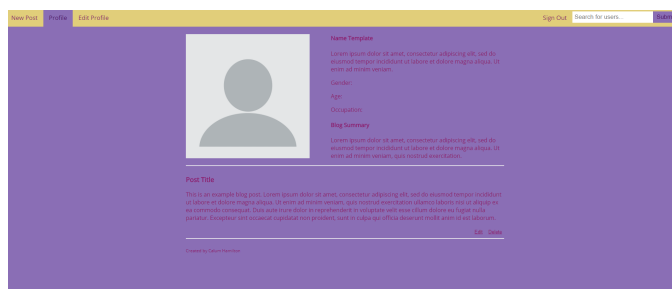Figure 2: **New Post Screenshot** -Screenshot of the New Post Page

Figure 3: **Profile Screenshot** -Screenshot of the Profile Page

4 Critical Evaluation

Evaluation Overall the design of the website matches the document requirements. The design of the page is
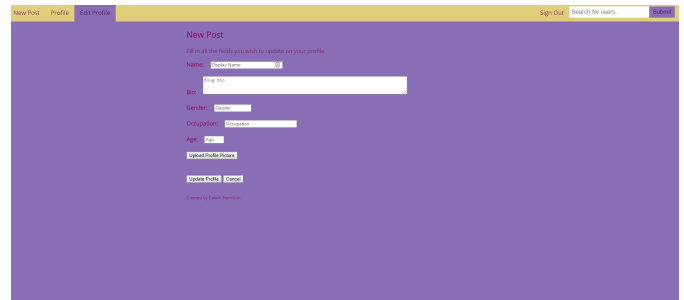

Figure 4: **Update Profile Screenshot** -Screenshot of the Profile Update Page

very clean, easy to navigate and easy to use. As a standard, the website is very consistent as well for usability. The spacing and layout is a lot better than the first coursework that I submitted. There is a clean standard navigation and color scheme is throughout the website. The index page however needs a restructuring to clean up the page and provide more clarity upon entering the website. I also needed more time to fully understand the potential of the different modules that I was using.

Next Time If I were to do the site again I'd give myself more time and have a full structure planned by the site beforehand. I'd get the login, sign up and search sorted and clean up a lot of the code. I'd also ensure that the pictures dimensions were very close to ensure that each of the cipher pages were as close as possible, to try and perpetuate good website practice.

5 Personal Evaluation

Learning I've learnt the difficulties with client and server-side communication. I've also gained a lot of experience downloading and installing modules. I've gained more experience programming on multiple levels and understand the distinction between server and client a lot more clearly now. I've also seen that it can be easy to make a website look really nice but actually getting it to work can be another matter altogether.

Challenges One of the biggest challenges was balancing workloads. Again, the Software Development Methods coursework was due the same day, and I was unable to balance my work very efficiently. This led to a lot of features being underdeveloped and unfinished, such as the three features mentioned above; search bar, login and sign up forms. I left them in however to show what the site would eventually look like with more time and development, as well as showing that there's minimal functions on the website, because I ran out of time.

I've faced a large number of challenges in the project, getting the server to run and work and getting communication working between the server and client side. Getting the JSON to save was also a very difficult challenge that I faced as well as automatically generating HTML. I dealt with these problems by changing the feature I was work-

ing on, and coming back to each feature one by one, to try and push the development cycle forward, even if there were parts where I was stuck.

Overall I feel like the project has taught me a lot about time management, web design techniques, server-side computing and communication via different protocols.

6 Conclusion

References

[1] https://pixabay.com, "Stock profile image,"

[2] https://jquery.com/, "Jquery module,"

[3] https://expressjs.com/, "Express module,"

[4] https://nodejs.org/en/, "Node.js module,"